Problem Set #6 (Full Version)

Greg Plaxton

April 27, 2019

Your solutions are required to be typeset using LaTeX, and to be turned in electronically using Canvas. As discussed on the course syllabus, there are two deadlines for this homework. The first deadline is 8pm on Friday, May 3. The second deadline is 8pm on Sunday, May 5. Submissions received before the first deadline will not be charged any slip days. No submissions will be allowed after the second deadline. See the syllabus for further details regarding the slip day policy.

1. Recall the load balancing problem that we discussed in class. We studied two greedy approximation algorithms for this problem. In the first of these algorithms, we processed the items in arbitrary order. In the second, we processed the items in nonincreasing order of weight. In class, we sketched a proof that the second algorithm achieves an approximation ratio of $\frac{4}{3}$. Prove that this bound is tight, in the following sense: For any $\varepsilon > 0$, there is an instance of the load balancing problem for which the approximation ratio achieved by the second algorithm exceeds $\frac{4}{3} - \varepsilon$.

2. Problem 35–2, page 1134.

3. In the lecture related to the weighted set cover problem, we presented a 0-1 ILP formulation of the problem, along with an associated LP relaxation. We constructed a family of set cover instances based on $\mathbb{F}_2^k$. We used linear algebra to analyze this family of instances and establish an $\Omega(\log n)$ bound on the integrality gap. In this problem, we will use a simpler randomized construction to establish the same $\Omega(\log n)$ bound.

   Let S be a set of n items. For any probability p (which may depend on n), we can construct a random subset of S using n independent flips of a p-biased coin: If the ith flip comes up "heads", then we include the ith item in the subset. Let S_p denote the corresponding probability distribution over subsets of S.

   We construct a random (unweighted) set cover instance $I_n = (S, \mathcal{F})$ as follows. For a suitable choice of the parameter ℓ (which may depend on n), we construct $\mathcal{F}$ by independently selecting ℓ sets from the distribution S_p.

   Prove that for a suitable choice of the parameters p and ℓ, there is a positive probability that both of the following conditions hold: (1) the optimal fractional objective function value for I_n is $O(1)$; (2) the optimal integral objective function value for I_n is $\Omega(\log n)$. (Hint: Use a Chernoff bound argument to prove that both conditions hold with high probability for your choices of p and ℓ.)

4. In the lecture on hypercube routing, we studied the "permutation routing problem" in which each node is the source and destination of exactly one packet. Consider a d-dimensional hypercube with $n = 2^d$ nodes. We saw that one way to route the packets to their destinations is to "correct bits" from left to right. For example, if $d = 8$ and we wish to route a packet from node 10100110 to node 11101101, then we send this packet along the path that visits the nodes 10100110, 11100110, 11101110, 11101100, 11101101. Clearly, this "left-to-right bit-fixing" scheme can be used to route a single packet from an arbitrary source to an arbitrary destination in at most d steps. On the other hand, we showed that the worst-case performance of this scheme for permutation routing is exponentially worse than d by presenting an $\Omega(\sqrt{n/d})$ lower bound. This negative result motivated us to study Valiant's two-phase permutation routing scheme. Under this scheme, the packets are routed to random intermediate destinations (via left-to-right bit-fixing) in the first phase, and then are routed to their final destination (again via left-to-right bit-fixing) in the second phase. We showed that Valiant's scheme routes an arbitrary permutation in $O(d)$ steps with high probability.

   A drawback of Valiant's scheme is that it achieves asymptotically optimal worst-case performance at the expense of increasing the number of hops traversed by the packets. For example, consider a packet being sent from an arbitrary source node to a uniformly random destination node. Under the left-to-right bit-fixing scheme, the expected number of hops traversed by the packet is $d/2$. On the other hand, under Valiant's two-phase scheme, the expected number of hops doubles to d. It is natural to ask whether there is a permutation routing scheme that achieves $O(d)$ worst-case performance with high probability without incurring this factor of two penalty.

   In this question, we analyze the following natural "randomized bit-fixing" scheme: To route a packet from a given source node y to a destination node y, we repeatedly correct a uniformly random incorrect bit. For instance, let us to revisit the example of routing a packet from node 10100110 to node 11101101. At the first step, we notice that four bits need to be corrected, so we pick one of these four uniformly at random and correct it. At the second step, three bits still need to be corrected, so we correct one of these three uniformly at random, et cetera. Notice that this scheme is equivalent to selecting a uniformly random shortest path from the source to the destination. To route a permutation using this scheme, we proceed as follows. First, each source independently selects a random bit-fixing path to send its packet to its destination (i.e., a random shortest path). Next, all of the packets are routed along their chosen paths to their destinations. As in the case of the other packet routing schemes that we have studied, packets can be delayed due to edge contention. Recall that we treat a hypercube edge (u, v) as a pair of independent directed edges (u, v) and (v, u), each of which can transmit a single packet in each time step, and each of which has an unbounded buffer for storing any packets that are waiting to traverse the edge.

Prove that there is a permutation π and a positive constant ε such that (no matter what queueing strategy is used to manage the edge queues) the expected time required for the random bit-fixing scheme to route π is $\Omega(n^{\varepsilon})$.